NFS-Root-Client Mini-Howto

par *Ofer Maor* , *ofer@hadar.co.il* Version 4.1, le 02 Février 1999

(Adaptation française par Antoine Levavasseur `levavass@efrei.fr`, le 4 mars 1999). L'objectif de ce mini-howto est d'expliquer comment créer l'arborescence racine d'un client pour le serveur qui gère des clients utilisant NFS-Root. La dernière version de ce mini Howto peut être récupérée à : *http://www.hcs.co.il/oferm/NFS-Root-Client/*

1 Copyright

Si vous avez des questions, contactez alors *Ofer Maor (* ofer@hadar.co.il *)*, l'auteur de ce Mini-Howto, ou *Greg Hankins*, le coordinateur des Howto pour **Linux**, *gregh@sunsite.unc.edu* par courrier électronique ou au +1 404 853 9989.

Si vous avez quelque chose à ajouter à ce Mini-Howto, envoyez par courrier électronique à l'auteur (Ofer Maor (*ofer@hadar.co.il*) vos informations. Toute information intéressante sera appréciée.

1.1 Remerciements

Je tiens à remercier l'auteur du Howto NFS-Root, Andreas Kostyrca (*andreas@medman.ag.or.at*). Son Mini-Howto m'a aidé lors de mes premiers pas pour mettre en place un client avec le système NFS-Root. Mon Mini-Howto ne cherche en aucun cas à remplacer ce travail, mais à l'améliorer grâce à ma propre expérience.

J'aimerais également remercier Mark Kushinsky (*marc026@ibm.net*) d'avoir relu et corriger les fautes de ce Howto, le rendant ainsi bien plus lisible.

2 Préface

Ce Mini-Howto à été écrit dans le but d'aider les gens qui veulent utiliser le montage par NFS-Root pour créer l'arborecence de leurs clients. Vous pourrez ainsi remarquer qu'il y a de nombreuses façon d'accomplir cela selon vos besoins et vos intentions. Si les clients sont individuels, et qu'ils ont tous leurs propres utilisateurs et administrateurs, il serai nécessaire de ne pas partager avec d'autres clients une partie significative des répertoires. Dans le cas ou le client est destiné à de multiples utilisateurs, et qu'ils sont tous administrés par la même personne (par exemple les ordinateurs d'une école), il faudrait partager le maximum de fichier afin de simplifier l'administration. Ce Howto, se concentre sur la deuxième possibilité.

2.1 Présentation Générale

Lorsque l'on construit le répertoire racine d'un client, en essayant de nous limiter à la taille minimale, nous chercherons principalement quels fichiers nous pouvons partager ou monter à partir du serveur. Dans ce Howto, je recommendrai la configuration d'un client basé sur ma propre expérience. Mais avant de commencer, veuillez remarquer que :

- Ce Mini-Howto n'explique pas comment effectuer le montage de partition par NFS-Root. Se référer au mini HOWTO NFS-Root si vous avez besoin de plus d'informations à ce sujet.
- J'ai basé la configuration de la plupart de mes clients sur les montages et les liens symboliques. Bon nombre de ces liens symboliques peuvent être remplacés par des liens durs. Chacun choisira selon ses préférences personnelles. Mettre un lien dur à travers un fichier monté a ses avantages, mais peut prêter à confusion. Un fichier ne sera pas effacé tant que tous ses liens durs ne seront pas effacés. Ainsi, pour prévenir les cas ou vous mettez à jour un certain fichier, et que les liens durs référencent toujours l'ancienne version, vous devez être très attentifs et garder une trace de tous les liens que vous mettez.
- Lorsque vous montez les informations à partir du serveur, il existe 2 possibilités. La première (la plus courante), est de monter la totalité du répertoire racine du serveur dans un répertoire local, et de simplement changer le chemin ou mettre des liens vers les répertoires correspondant. Personnellement, je n'aime pas monter la partition racine du serveur sur les clients. Ainsi, ce Howto suggère une façon de monter les répertoires appropriés du serveur aux endroits appropriés du système.
- Ce Howto est basé sur mon expérience à construire les répertoires clients sur la distribution Slackware 3.1. Certaines choses peuvent être différentes (en particulier les fichiers rc.*), pour d'autres utilisateurs, cependant les concepts restent les mêmes.

3 Créer le répertoire racine du client

3.1 Création de l'arborescence

Tout d'abord, vous devez créer la structure des répertoires elle-même. J'ai créé tous mes clients dans le répertoire `/clients/hostname` et je l'utilise pour tous les exemples suivants. Cela peut toutefois être remplacé par n'importe quoi d'autre. La première étape, est donc de créer les répertoires appropriés sur la racine. Vous devez créer les répertoires suivants :

```
bin, dev, etc, home, lib, mnt, proc, sbin, serveur, tmp, usr, var
```

et tous les autres répertoires que vous souhaitez avoir sur votre système.

Les répertoires locaux `proc` et `dev` doivent être distincts pour chaque machine alors que le reste des répertoires peut être soit en partie soit complètement partagé avec les autres clients.

3.2 Création d'un système de fichier minimal pour le boot

3.2.1 Créer le répertoire dev.

Bien que le répertoire `dev` puisse être partagé, il est préférable d'en créer un différent pour chaque client. Vous pouvez créer le répertoire dev de votre client avec le script MAKEDEV approprié, cependant dans la plupart des cas, il est plus simple de le recopier à partir du serveur :

```
cp -a /dev /clients/hostname
```

Vous devez garder à l'esprit que `/dev/mouse`, `/dev/cdrom` et `/dev/modem` sont des liens vers les périphériques courants, il faut donc s'assurer qu'ils sont correctement reliés pour correspondre au matériel du client.

3.2.2 Copier les exécutables indispensables.

Bien que nous montons tout à partir du serveur, il y a un minimum que vous devez recopier pour chaque client. Avant tout, vous avez besoin de "init", car votre système ne sera pas capable de faire marcher quoi que ce soit avant de s'être initialisé (l'auteur l'a difficilement trouvé ;-). Donc premièrement, vous devez copier `/sbin/init` dans le répertoire sbin de votre client ensuite, pour que le fichier rc.S puisse être lancé, vous devez copier /bin/sh dans le répertoire binaire de votre client. Ensuite, de manière à monter tout ce dont vous avez besoin, vous devez copier `/sbin/mount` dans le répertoire sbin de votre client. C'est le minimum, à partir du moment ou la première ligne de votre rc.S *(rc.sysinit pour une distribution redhat)* est `mount -av`.

Cependant, je recommande de copier quelques autres fichiers : `update, ls, rm, cp et umount` comme cela, vous aurez les outils minimum au cas ou le client à des problèmes de montage. Si vous choisissez d'utiliser votre swap avant de monter la racine, vous devez également copier l'exécutable `swapon`.

Bien que la plupart de ces exécutables ont eu une édition des liens dynamique, vous devez aussi copier une petite partie de /lib :

```
cp -a /lib/ld.* /lib/libc.* /lib/libcurses.* /client/hostname/lib
```

N.d.T. : sur certaines distributions (en particulier RedHat) ce sont les fichiers /lib/libtermcap qu'il faut recopier à la place de /lib/libcurses.* selon la façon dont sont compilés les exécutables.*

Mettre des liens durs vers les fichiers eux-mêmes, au lieu de les copier, est une solution à considérer. Relisez mes commentaires à ce sujet dans la partie 2.1 de ce Howto.

Veuillez remarquer que toutes les informations précédentes supposent que le noyau ai obtenu tous les paramètres lors du boot. Si vous pensez utiliser **rarp** ou **bootp**, vous aurez probablement besoin des fichiers qui leurs correspondent.

Généralement, vous aurez besoin du minimum de fichiers qui vous permettent de configurer le réseau et d'exécuter rc.S *(ou rc.sysinit sur Redhat)* jusqu'au moment ou est monté le reste du système de fichier. Assurez-vous que vous avez vérifixiée vos fichiers `/etc/init` et `rc.S`, vérifiez qu'il n'y a pas de "surprises" ou quoi que ce soit d'autre qui aurait besoin d'accéder à d'autres fichiers, avant que le premier montage ait lieu. Si vous le faites toutefois, retrouvez ces fichiers, vous pouvez soit les recopier également, soit enlever les parties correspondantes de vos fichiers `init` ou `rc.S`.

3.2.3 Le répertoire var

Le répertoire var, dans la plupart des cas, doit être séparé pour chaque client. Cependant, bon nombre des données peuvent êtres partagées. Créez dans le répertoire `/serveur`, un répertoire appelé `var`. Nous recopierons le répertoire `var` du serveur à cet endroit. Pour créer le répertoire `var` local, tapez simplement :

```
cp -a /var /clients/hostname
```

Maintenant, vous pouvez choisir ce que vous voulez séparer, et ce que vous voulez partager. Pour tout répertoire/fichier que vous voulez partager, enlevez-le simplement du répertoire var du client, et mettez un lien symbolique vers le répertoire `/serveur/var`. Cependant, vous remarquerez que vous pouvez mettre un lien symbolique vers `/serveur/var` ou vers `../serveur/var` mais surtout PAS vers `/clients/hostname/serveur/var`puisque cela ne marchera plus lorsque la racine aura changé.

Généralement, je recommande de séparer `/var/run, /var/lock, /var/spool` et `/var/log`.

3.2.4 Le cas des autres répertoires

- etc est détaillé dans la section suivante.
- mnt et proc sont utilisés localement
- usr et home sont simplement des points de montage
- tmp est à votre bon choix. Vous pouvez créer un répertoire différent pour chaque client, ou créer un répertoire `/clients/tmp`, et le monter sur chaque client dans `/tmp`. Je recommande de fournir un répertoire tmp pour chaque client

3.3 Construire le répertoire etc et configurer le client

Remarque - cette section traite de la façon de créer un répertoire `etc` qui est principalement séparé entre les clients. Si votre station diskless a des administrateurs systèmes différents, il est préférable de faire un répertoire `etc` séparé pour chaque client.

3.3.1 Mettre un grand répertoire etc

Bien que nous séparions les répertoires `etc` des clients, nous souhaitons toujours partager une bonne partie des fichiers de cet endroit. D'une façon générale, je pense que partager les fichiers de etc avec le répertoire `/etc` du serveur est une mauvaise idée, et c'est pourquoi je recommande de créer un répertoire `/clients/etc`, qui contiendra les informations dont les clients ont besoin. Pour commencer, recopier simplement le contenu de /etc du serveur vers le répertoire `/client/etc`.

Vous devrez ajouter à ce répertoire tous les fichiers de configuration qui ne dépendent pas de la machine, par exemple motd, issue, etc. mais aucun de ceux dépendant du client (i.e. inittab ou fstab)

Les changements les plus importants doivent être faits dans le répertoire `rc.d`. Premièrement, vous devez changer rc.inet1 pour qu'il corresponde à votre paramétrage local. Je passe tous mes paramètres réseau au noyau avec LILO/Loadlin, c'est pourquoi j'ai enlevé presque tout de mon fichier `rc.inet1`. La seule chose que j'ai laissé, est le **ifconfig** et la **route** vers l'entrée localhost. Si vous utilisez rarp ou bootp, vous devez en tenir compte lors de la création du fichier.

Deuxièmement, vous devez éditer le fichier rc.S. D'abord, enlever toutes les parties qui sont responsable d'une recherche d'erreur avec fsck, puisque fsck aura lieu alors que le serveur est déjà démarré. Ensuite, vous devez trouver la ligne ou est monté votre fstab. Cela doit ressembler à :

```
mount -avt nonfs
```

Le **-t nonfs**. est jusque là normal puisque les client lancent en premier rc.S et seulement après rc.inet1 pour configurer l'Ethernet. Mais comme cette ligne ne va pas monter les partitions NFS, elle doit être détruite. Il faut ensuite la remplacer par un `mount -av` . Si vous avez besoin d'utiliser rarp/bootp pour configurer votre réseau, faites le dans rc.S (ou appelez le script approprié dans rc.S), avant de monter, et assurez vous que les répertoires bin et sbin contiennent les fichiers nécessaires.

Après que le `mount -av` ait été accompli, vous devez avoir un système de fichier opérationnel. Créez un fstab générique, que vous pourrez recopier sur chaque client.

Votre `fstab` doit ressembler à :

```
serveur:/clients/hostname    /              nfs     default  1 1
serveur:/bin                 /bin           nfs     default  1 1
serveur:/usr                 /usr           nfs     default  1 1
serveur:/sbin                /sbin          nfs     default  1 1
serveur:/home                /home          nfs     default  1 1
serveur:/lib                 /lib           nfs     default  1 1
serveur:/clients/etc         /serveur/etc   nfs     default  1 1
serveur:/clients/var         /serveur/var   nfs     default  1 1
none                         /proc          proc    default  1 1
```

Veuillez remarquer, que le mot clé _default_ ne doit pas fonctionner avec toutes les versions de mount. Vous devrez alors le changer en _rw_ ou _ro_ ou enlever toute la partie _default 1 1_ .

Assurez vous également que sur votre serveur le `/etc/exports` ressemble à :

```
/clients/hostname   hostname.domainname(rw,no_root_squash)
/clients/etc        hostname.domainname(ro,no_root_squash)
/clients/var        hostname.domainname(ro,no_root_squash)
/usr                hostname.domainname(ro,no_root_squash)
/sbin               hostname.domainname(ro,no_root_squash)
/bin                hostname.domainname(ro,no_root_squash)
/lib                hostname.domainname(ro,no_root_squash)
/home               hostname.domainname(rw,no_root_squash)
```

Mise à part la première ligne qui doit être différente pour chaque client, les autres lignes peuvent être remplacées par un masque pour s'appliquer à tout vos clients (comme pc*.domain - gardez à l'esprit, que * se substitue uniquement aux chaînes de caractères ne contenant pas de point). Je propose de mettre la majorité des répertoires en lecture seule *ro*, mais c'est à vous de choisir. Le `no_root_squash` assure que le super-utilisateur aura sur les clients les permissions correspondantes avec nfsd. Jetez un coup d'oeil au `man exports(5)`. Si vous souhaitez que vos utilisateurs puissent utiliser **passwd** également à partir des clients, assurez-vous que /etc ai les permissions rw et non ro. Toutefois faites comme vous le souhaitez.

Vouz remarquerez aussi, en ce qui concerne le fichier rc.S que sur la Slackware, par défaut, un fichier `/etc/issue` et `/etc/motd` sont créés à chaque exécution. Cette fonction DOIT être désactivée si ces fichiers sont montés en ro à partir du serveur, je recommande de les désactiver dans tous les cas.

Pour terminer, si vous voulez avoir une base d'utilisateurs commune à votre serveur et vos clients, vous pouvez choisir entre 1), utiliser NIS (lire le NIS-Howto), et ainsi, chaque client aura un `/etc/passwd` et `/etc/group` séparé puisqu'il les recevra du serveur. 2) Dans la plupart des cas, un lien symbolique suffira. Pour cela, vous pourrez soit mettre un lien dur de `/client/etc/passwd` vers `/etc/passwd`, ou si vous préférez un lien symbolique, reliant `/etc/passwd` vers `/client/etc/passwd` (et pas dans l'autre sens puisque le client ne monte pas le répertoire etc du serveur). Faire de même avec `/etc/group`

3.3.2 Créer le répertoire etc du client

Généralement, la plupart des fichiers du répertoire etc du client doivent avoir des liens symboliques vers `/serveur/etc`. Toutefois, certains fichiers sont différents pour chaque machine, et certains doivent être là lorsque le noyau est chargé. Le minimum dont vous avez besoin dans le répertoire etc est :

```
resolv.conf
hosts
inittab
rc.d/rc.S
fstab
```

Puisque ces 5 fichiers peuvent êtres identiques sur les clients, vous pouvez simplement les recopier ou mettre des liens durs. Cependant, il est judicieux de garder les fichiers `fstab` et `rc.S` différents pour chaque client. Vous devez également séparer `etc/HOSTAME` pour chaque client. Personnellement, je recommande d'avoir les fichiers de `rc.d` distincts pour chaque client puisque la configuration et le matériel peuvent varier d'une machine à l'autre.

Pour chaque client, ajouter dans la `fstab` la ligne swap appropriée :

```
/dev/swap_partition                            swap     swap   default 1 1
```

Pour le reste des fichiers de /etc du client, vous pouvez soit mettre des liens durs vers `/clients/etc/*` , ou mettre des liens symboliques vers `/serveur/etc` (qui est l'endroit où est monté `/clients/etc`).

Assurez-vous que votre machine peut résoudre correctement, soit par **named** ou grâce à **etc/hosts**. Ce n'est pas une mauvaise idée de garder l'adresse IP du serveur dans `/etc/hosts`, plutôt que de compter sur la résolution. Si vous comptez uniquement sur la résolution de noms, un problème avec `named` empêchera vos clients de booter.

3.4 Booter la machine

Maintenant que vous avez fait tout ce qu'il faut pour démarrer votre machine, croisez vos doigts et priez pour que tout marche comme il faut :-)

4 Créer d'autres clients

Si vous avez suivi mes instructions jusqu'ici, cela devrait être facile - changer de répertoire vers `/clients` et tapez :

```
cp -a hostname1 hostname2
```

et vérifiez bien les points suivants :

> Les fichiers rc.d/* correspondent au matériel et la configuration logicielle **etc/HOSTNAME** est correcte, la ligne de swap de fstab est correcte, les liens symboliques `dev/mouse, dev/modem` et `/dev/cdrom` sont corrects.

`Bonne Chance...`